

The Effect of Interaction Granularity on Learning with a Data Normalization Tutor

Amali Weerasinghe, Antonija Mitrovic, Amir Shareghi Najar and Jay Holland

Intelligent Computer Tutoring Group
University of Canterbury, Christchurch, New Zealand
{amali.weerasinghe;amir.shareghinajar}@pg.canterbury.ac.nz
{tanja.mitrovic;jay.holland}@canterbury.ac.nz

Abstract: Intelligent Tutoring Systems (ITSs) have proven their effectiveness in many instructional domains, ranging in the complexity of domain theories and tasks students are to perform. The typical effect sizes achieved by ITSs are around 1SD, which are still low in comparison to the effectiveness of expert human tutors. Recently there have been several analyses done in order to identify the factors that contribute to success of human tutors, and to replicate it in ITSs. VanLehn [6] proposes that the crucial factor is the *granularity of interaction*: the lower the level of discussions between the (human or artificial) tutor and the student, the higher the effectiveness. We investigated the effect of interaction granularity in the context of NORMIT, a constraint-based tutor that teaches data normalization. Our study compared the standard version of NORMIT, which provided hints in response to errors, to a version which used adaptive tutorial dialogues instead. The results show that the interaction granularity hypothesis holds in our experimental situation, and that the effect size achieved is consistent with other reported studies of a similar nature.

Keywords: effectiveness of ITSs, interaction granularity hypothesis, empirical study, tutorial dialogues, NORMIT

1 Introduction

One-to-one human tutoring is widely considered to be the most effective form of tutoring. Students' learning gains increase by two standard deviations when tutored by expert human tutors compared to traditional classroom instruction [1]. Researchers have been trying to identify the factors that contribute to the success of human tutors, and replicate it in ITSs. One of the frequently discussed factors is interactivity, since human tutoring is highly interactive. ITSs are also interactive: typically students are engaged in problem solving, and receive guidance in the form of feedback, adaptive problem selection and other interventions.

Many questions related to interactivity have been posed in the ITS literature. Koedinger and Aleven [2] investigate the assistance dilemma, which refers to the problem of balancing assistance giving and withholding in order to optimise student learning. Information might be provided to the student; for example, the student might be given advice how to solve a particular problem step. On the other hand, infor-

mation might be elicited: the student might be asked a series of questions leading to the correct action. Similarly, worked-out examples might be given to the student, but if the student is prompted to self-explain the examples to him/herself, learning is greatly enhanced [3-5]. Cognitive tutors, for example, provide immediate positive and negative feedback on each step, instructions on how to complete the step on demand, adaptive problem selection and other forms of interactive guidance.

In a recent paper, VanLehn [6] examines several hypotheses that aim to explain the effectiveness of human tutoring and concludes that only two provide viable explanations. Human tutors provide frequent feedback which allows students to repair their knowledge. They also provide adaptive scaffolding in terms of tutorial dialogues. VanLehn proposes that the crucial factor for the effectiveness of instruction is the *interaction granularity*. Human tutoring has no limitation on interaction granularity as human tutors intervene very frequently and also at various levels. On the other extreme, if there is *no tutoring* provided whatsoever, the student is solving problems with no feedback and needs to do a lot of reasoning which often is unproductive because of lack of knowledge. In between those two extremes, VanLehn discussed three types of computer-based tutoring. In *answer-based tutoring* (such as in Computer-Aided Instruction), the student only submits the final answer for the problem without intermediate steps and therefore only limited feedback can be given. ITSs typically provide *step-based tutoring*, as they react (or may react) on each step of the solution, which represents a finer grain size. Dialogue-based ITSs represent *sub-step tutoring*, and are characterised with an even finer level of interaction granularity compared to step-based ITSs. Tutorial dialogues allow the ITS to obtain more information about the student's reasoning in comparison to step-based tutoring.

The interaction granularity hypothesis predicts that the effectiveness of tutoring increases as the granularity of interaction decreases (i.e. the grain size becomes smaller). VanLehn conducted a large meta-review of reported studies, each of which compared two instructional conditions that differ in the interaction granularity level keeping other factors the same. The meta-review confirmed the hypothesis, with the limitation that in some types of comparisons the number of studies was small.

Using the interaction granularity criterion, the studies we have performed can be classified into several groups. Some studies compared step-based to answer-based tutoring (e.g. [7]) or no tutoring [8]. We also performed studies comparing various forms of step-based tutoring to one another [9], or various forms of sub-step tutoring [10]. In this paper we report on a study that compares step-based to sub-step tutoring. Section 2 presents the step-based constraint-based tutor for data normalization, while Section 3 discusses the substep-based version of the same ITS. We then present the design of our study in Section 4, followed by the results in Section 5 and conclusions.

2 NORMIT

NORMIT [5, 11] is a constraint-based tutor that teaches data normalization, a technique which consists of refining an existing relational database schema in order to ensure that all relations are of high quality [12]. Normalization is a hard topic for

students [5, 13], as it requires theoretical knowledge of the relational data model, functional dependencies (FDs), normal forms and the related algorithms.

Data normalization is a procedural technique, consisting of a sequence of tasks to analyze the quality of a database. Each problem consists of a relation schema and a set of given FDs. For example, problem 13 is defined on relation $R(A, B, C, D, E)$ (typically the semantics of the attributes is not given) and the set of FDs: $\{A \rightarrow BC, CD \rightarrow E, AC \rightarrow E, B \rightarrow D, E \rightarrow AB\}$ (see Figure 1).

Fig. 1. A screenshot of NORMIT

The normalization procedure as implemented in NORMIT consists of eleven tasks described below. Please note that we refer to elements of the procedure as *tasks* rather than *steps*, as each of them contains a number of actions the student has to perform, including in some cases relatively complex algorithms. The first eight tasks are necessary to determine the highest normal form the relation is in. If the relation is not in Boyce-Codd Normal Form (BCNF), the student needs to apply the relational synthesis algorithm to derive an improved database schema via tasks 9-11.

1. Identify the candidate keys for the given table. There may be one or more keys in a table; e.g. in problem 13 there are four candidate keys: A, E, BC and CD.
2. Find the closure of a given set of attributes. For example, to make sure that E is a candidate key, the student can check that closure contains all attributes of R.
3. Identify prime attributes. Prime attributes are those attributes that belong to any candidate key. In problem 13, all attributes are prime.
4. Simplify FDs by applying the decomposition rule, if necessary. For example, $A \rightarrow BC$ is replaced with two FDs: $A \rightarrow B$ and $A \rightarrow C$.
5. Determine the normal forms for the given relation.

6. If the student specified that the relation is not in 2NF, he/she needs to identify FDs that violate that form (i.e. partial FDs).
7. If the student specified that the relation is not in 3NF, he/she needs to identify FDs that violate that form (i.e. transitive FDs).
8. If the student specified that the relation is not in BCNF, he/she will be asked to identify FDs that violate that form.
9. For relations that are not in BCNF, reduce LHS of FDs. This task checks whether some of the attributes on the LHS can be dropped while still having a valid FD.
10. Find minimal cover (i.e. the minimal set of FDs).
11. Decompose the table by using the minimal cover.

NORMIT teaches data normalization in a task-by-task manner, showing only one task at a time which the student needs to complete before moving on to the next task. Figure 1, for example, shows the candidate keys task of problem 13. The student can submit a solution at any time, which the system then analyses and presents feedback. At any point during the session, the student may change the problem, review the history of the session, examine the student model or ask for the full solution. The system currently contains 50 problems and new problems can be added easily. NORMIT is a constraint-based tutor, and its knowledge base is represented as a set of 82 (problem-independent) constraints. Each constraint is relevant for a particular task of the procedure. Some constraints are purely syntactic, while others compare the student's solution to the ideal solution (generated by the problem solver). The short-term student model consists of a list of violated/satisfied constraints for the current attempt, while the long term model records the history of usage for each constraint.

3 The Model for Adaptive Tutorial Dialogues

In previous work [14] we developed a general model of adaptive tutorial dialogues, which we used to provide tutorial dialogues in NORMIT. This model consists of three parts: an error hierarchy, a set of tutorial dialogues and rules for adapting them. The error hierarchy categorises all error types in a particular domain. Each leaf in the hierarchy is associated with one or more violated constraints, which are covered by a single tutorial dialogue. The error types are grouped into higher-level categories, with the top three levels of the error hierarchy being domain-independent. At the top level of the hierarchy, errors are classified into syntax or semantic ones. Semantic errors are further classified into several groups, such as missing components or extra components in the solution.

The student model is extended with the information about the errors the student made during interaction. This new component of the student model stores the frequency of the student making a mistake corresponding to each node of the error hierarchy. When a student submits a solution to the current problem, a set of violated constraints (if any) is determined. The information about violated constraints is then used to update the violation frequencies for the relevant nodes in the hierarchy. If the student's solution contains several errors, the most suitable error for discussion is

selected from the error hierarchy. The model identifies the error that was most frequent and the corresponding dialogue is then used for discussion with the student.

Tutorial dialogues (the second component of our model of tutorial dialogues), are hand-crafted for each error type. For syntax errors, dialogues are very simple, and consist of a single feedback message (the same message used as the hint¹ in the original version of NORMIT). For semantic errors, tutorial dialogues consist of four levels of prompts. Each prompt contains a question and a set of three options for the student to respond. For example, the tutorial dialogue in Figure 2 is used when the solution contains an incorrect candidate key, such as AD for problem 13. The first-level prompt (*NORMIT1*) presents a conceptual question. If the student fails to answer correctly, the model poses another question at the same level (“*What do we mean by a candidate key being a minimal set of attributes?*”). The tutor reveals the correct answer if the student cannot identify it.

The second level presents a *reflective prompt*, asking why the solution is incorrect (*NORMIT2*, shown in Figure 1). If the student fails to answer correctly, the tutor makes the reasoning process more explicit. In this scenario, the tutor says “*Let’s see ... Check carefully whether all the attributes of AD are needed to determine all the other attributes*”. The student needs to press the Continue button to proceed, after which the next prompt is revealed: “*Do we need all the attributes of AD to determine all the other attributes?*”

The third level provides a *corrective action prompt* which helps the student to correct the error (*NORMIT3*). Finally, the *conceptual reinforcement prompt* provides another opportunity to learn the corresponding domain concept (*NORMIT4*). This level uses different types of questions such as simple questions, fill-in-the-blank (*NORMIT4*) or true-false questions, to keep the student engaged. The first and last levels focus on problem-independent details and support the acquisition of conceptual knowledge. On the other hand, the second and third levels discuss problem-specific details.

NORMIT1: You seem to be having some difficulty with candidate keys. Let's look at candidate keys in detail. What is a candidate key?

Student1: A candidate key is a minimal set of attributes that determines all the attributes

NORMIT2: Great, can you tell me what's wrong with AD being a candidate key?

Student 2: It is not a minimal key

NORMIT3: Good job! Now tell me the extra attributes for AD?

Student3: D

NORMIT4: Well done! Now try to answer this question so that you'll remember what you learnt.
A is akey with extra

Student4: superkey, candidate key, attributes

NORMIT5: Great job! Please make the changes and submit the solution when you're done.

Fig. 2. An example tutorial dialogue

The last component of the model of tutorial dialogues is adaptation rules, which enable individualization of the selected dialogue. The rules use the student model and

¹ NORMIT offers six levels of feedback [5]

the session history to decide on the timing, selection and the dialogue content. The number of prompts a student receives depends on the accuracy of his/her answer and the number of times an error is made. If an error is made for the first time, a student receives a reflective prompt. If answered correctly (as in the situation illustrated in Figure 1), the student is allowed to resume problem solving. If the same error is made two times, the student receives two prompts: reflective and corrective action prompt. If the same error is repeated three times or more, the student must go through the entire dialogue starting from the conceptual prompt. At any level, an incorrect answer will trigger another prompt to be presented as there is no evidence the student has the relevant knowledge. Adaptation rules also deal with situations when students abandon problems, or are inactive for a period of time. In such cases, the current state of the student's solution is evaluated and a dialogue is initiated.

Steps
FIND CANDIDATE KEYS · FIND PRIME ATTRIBUTES · SIMPLIFY FUNCTIONAL DEPENDENCIES · DETERMINING NORMAL FORM

Problem 13

Task: Find all candidate keys for the following table with the given functional dependencies.

Relation: R (A, B, C, D, E)

Functional Dependencies:
 $\{A\} \rightarrow \{B, C\}$
 $\{C, D\} \rightarrow \{E\}$
 $\{A, C\} \rightarrow \{E\}$
 $\{B\} \rightarrow \{D\}$
 $\{E\} \rightarrow \{A, B\}$

Feedback
A candidate key you specified is not minimal. You need to remove the extra attributes.
Continue

Enter each candidate key into the space provided (if the key contains more than one attribute each attribute should be separated by a comma):

Candidate keys: **Add** E **Delete**
A,D **Delete**

Feedback: **Submit Answer** **Show Full Solution** **Clear**
Compute Closure

Fig. 3. A screenshot of problem 13 for a participant in the Hint group²

4 Study

Our goal was to investigate the effect of interaction granularity on learning data normalization. We conducted a study in October 2012 at the University of Canterbury, involving volunteers from an introductory database course. The study was conducted during a regular lab session (100 minutes long) in the eleventh week of the course, by which time the students had already learnt about data normalization in lectures. Participants were randomly assigned to two groups. Error selection was done in the same way in both groups, as described in Section 3. The Dialogue group received adaptive tutorial dialogues in response to errors, while the Hint group received non-interactive

² The problem-solving area (left pane) was disabled when feedback/dialogues were presented

hint messages (Figure 3). In addition to hints/dialogues, both conditions received error-flag feedback: the incorrect part of the solution corresponding to the selected error was highlighted in red (AD in Figures 1 and 3). Both groups could also request the solution for the current task.

The Hint group received non-interactive hint messages. A hint message is attached to a constraint, and is provided as feedback when the constraint is violated. The theory of learning from performance errors [15] specifies that effective feedback should tell the user where the error is, what constitutes the error (perform blame allocation), and refer to the underlying domain concept. NORMIT highlights the incorrect part of the student's solution, while the feedback message specifies what is wrong. In Figure 3, the hint informs the student that the highlighted candidate key is not minimal.

The study consisted of three phases: pre-test, interaction and post-test. The tests were of similar complexities and consisted of four questions each. The first two questions focused on procedural knowledge whereas the remaining two were for conceptual knowledge.

5 Results

37 students participated in the study. Data about one participant was excluded, as the student spent only 5 minutes working with NORMIT, resulting in 18 students in each group. Some students have not completed the post-test. Table 1 reports the statistics for students who submitted both tests. The groups had similar performances on the pre- and post-test. Both groups improved significantly between the pre- and post-test. In the pre-test, we also asked the students about how interested they were in learning data normalisation, on the Likert scale from 1 (not interested at all) to 5 (very interested). The Mann-Whitney U-test revealed no significant differences between the two groups on this question.

Table 1. Performance of the students who submitted both tests

	Hint (17)	Dialogue (15)	p
Pre-test (%)	66.91 (25.36)	63.33 (22.89)	.34
Post-test (%)	82.35 (18.78)	89.17 (16.95)	.14
Improvement pre-to-post	t=-2.18, p=.022	t=-5.57, p<.01	
Gain	15.44 (29.16)	25.83 (17.97)	.11
Interest	3.35 (0.49)	3.13 (0.92)	.48

Table 2 provides additional statistics about the study. There was no difference in learning time, the number of attempted and solved problems, the total number of attempts (i.e. submissions) and learnt constraints. Learnt constraints are those that the student did not know at the beginning of the session, but learnt during the session. There was also no significant difference in the number of interventions (in the form of hints or adaptive dialogues) the two conditions received. The effect size (Cohen's *d*) based on the learning gain 0.42, which is a medium size effect.

Table 2. Basic statistics for all participants

	Hint (18)	Dialogue (18)	p
Time (min)	70.22 (21.06)	73.89 (13.72)	.27
Attempted problems	10.28 (5.21)	9.50 (4.62)	.32
Solved problems	9.33 (4.97)	8.28 (4.11)	.25
Total attempts	109.22 (62.10)	99.44 (44.73)	.29
Learnt constraints	3.28 (3.00)	4.05 (2.99)	.22
No. of hints/dialogues	35.67 (22.57)	33.89 (18.15)	.39

Table 3 presents some details about the dialogues. Approximately one third of the dialogues were single-level ones; in those situations the Dialogue group participants received the same feedback (i.e. hints) as their peers. The remaining dialogues were multi-level dialogues. In such cases, the students saw on average 7.11 dialogues with only one prompt (because they successfully answered the prompt, as explained in Section 3) and 14.84 multi-prompt dialogues. The average number of prompts in the multi-prompt dialogues was 2.74. The average success rate in answering prompts was 71%. The students received more problem-specific prompts (28.78) than problem-independent prompts (18.33), which was to be expected as that is the result of adaptation rules. The success rate on two types of prompts is comparable.

Table 3. Dialogue analyses

Single-level dialogues seen	11.89 (7.79)
Multi-level dialogues seen	22 (12.77)
Single-prompt dialogues	7.11 (4.01)
Multi-prompt dialogues	14.84 (9.57)
No of prompts in a multi-prompt dialogues	2.74 (0.30)
Total number of questions answered	47.11 (28.78)
% of prompts answered correctly	64.9 (17.61)
% of prompts answered incorrectly	19.9 (9.03)
% of prompts answered with a More Help request	15.2 (19.20)
Total number of problem-independent prompts	18.33 (13.69)
% of problem-independent prompts answered correctly	67.616 (17.68)
Total number of problem-specific prompts	28.78 (15.63)
% of problem-specific prompts answered correctly	64.29 (23.84)

We performed a finer analysis of the learning gains, looking at two types of questions (conceptual/procedural) in the tests (Table 4). There was no difference between the pre-test performances on both types of questions. The performance of the Dialogue group on the procedural questions is marginally significantly higher than that of the Hint group, but there is no significant difference on the gains. The Dialogue group improved significantly between pre and post-test on both types of questions, while the Hint group only improved significantly on conceptual questions. The effect size on procedural questions is 0.35, while the effect size for conceptual questions is 0.21. Therefore, tutorial dialogues enabled the students to significantly improve both conceptual and procedural knowledge, while the hints resulted in a significant improvement of conceptual knowledge only.

Table 4. Comparison of the two groups on two types of questions

Procedural questions	Hint (17)	Dialogue (15)	p
Pre-test (%)	64.71 (23.48)	63.33 (35.19)	p= .44
Post-test (%)	70.59 (30.92)	83.3 (24.40)	t = -1.30, p = .10
Improvement pre-to-post	p = .29	t = -2.102, p = .03	
Gain	5.88 (42.87)	20 (36.84)	p = .16
Conceptual questions	Hint (17)	Dialogue (15)	p
Pre-test (%)	69.12 (34.83)	63.33 (35.19)	p = .3
Post-test (%)	94.12 (14.06)	95 (14.02)	p = .43
Improvement pre-to-post	t = -3.36, p < .01	t = -3.67, p < .01	
Gain	25 (30.62)	31.67(33.36)	p = .28

6 Conclusions

The interaction granularity hypothesis states that the effectiveness of tutoring increases as the granularity of interaction decreases. We conducted a study comparing two versions of the data normalization tutor, a step-based and a substep-based version. The Hint group (providing step-based tutoring) received non-interactive messages on their errors, while their peers in the Dialogue group received adaptive, interactive dialogues that discuss their errors (substep-based tutoring). The same mechanism was used in both conditions to select an error for presenting feedback or an adaptive dialogue in the case of multiple errors in the student's submission. In other words, if two students with identical interaction histories submit identical solutions, the error selected for discussion will be the same although the two students are from two different conditions. The main difference between the two groups lies in the instructional intervention in response to the selected error. While the Hint group participant received a single feedback message for the error, the participant from the Dialogue group was engaged in an interactive, adaptive dialogue for the same error. Additionally, in both conditions the selected error was flagged, and both groups had access to the solution for the current task of the procedure. As the performances of both groups improved significantly from pre- to post-test, both interventions (hints and dialogues) assisted the students to acquire knowledge about data normalization.

Further analysis of the effect of interactions with NORMIT on acquiring conceptual and procedural knowledge revealed interesting results: (i) the Dialogue group improved significantly both on conceptual knowledge and procedural knowledge; (ii) the Hint group significantly improved only on conceptual knowledge. The effect size on the procedural knowledge gain is 0.35, while for conceptual knowledge it is 0.21. The differences in gains between the Dialogue and Hint groups are not statistically significant, due to the small size of the study, but the trends are consistent with the interaction granularity hypothesis. The Hint group was presented with a non-interactive feedback message about the selected error; the hints were pre-specified messages of conceptual nature. Hints discuss the underlying domain concepts that are relevant for the incorrect part of the student's solution (identified via error flagging). The students from the Hint condition, however, need to reason about feedback. They were not told explicitly how to correct the error, unless they accessed the solution for the current task.

On the other hand, the Dialogue group participants were engaged in a discussion about both relevant domain concepts and the problem-solving procedure. This condition received more scaffolding via adaptive dialogues. The dialogues approach the error from multiple aspects, such as why the student's solution is incorrect, how to correct it and corresponding domain knowledge. The student involved in a tutorial dialogue is more engaged than a student who receives a hint message. As dialogues were adaptive, the number of prompts depended on the history of the individual tutoring session. Therefore, the granularity of the interaction was significantly lower for the Dialogue group than the Hint group. The effect size of our study (0.42) is of the same magnitude as the effect sizes reported for studies of similar nature in [6], thus providing another supporting evidence for the interactivity granularity hypothesis.

References

1. Bloom, B.S. The 2-sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13, 4-16 (1984).
2. Koedinger, K., Aleven, V.: Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19, 239-264 (2007).
3. Chi, M.T.H., De Leeuw, N., Chiu, M.H., LaVancher, C.: Eliciting self-explanations improves understanding. *Cognitive science*, 18, 439-477 (1994).
4. Aleven, V., Koedinger, K.R.: An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science*, 26, 147-179 (2002).
5. Mitrovic, A.: The effect of explaining on learning: a case study with a data normalization tutor. In: C-K Looi, G. McCalla, B. Bredeweg, J. Breuker (eds) Proc. Artificial Intelligence in Education, IOS Press, pp. 499-506 (2005).
6. VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist*, 46(4), 197-221 (2011).
7. Suraweera, P., Mitrovic, A.: An Intelligent Tutoring System for Entity Relationship Modelling. *Artificial Intelligence in Education*, 14(3-4), 375-417 (2004).
8. Amalathas, S., Mitrovic, A., Ravan, S.: Decision-making tutor: providing on-the-job training for oil palm plantation managers. *Research and Practice in Technology Enhanced Learning*, 7(3), 131-152 (2012).
9. Mitrovic, A. Fifteen years of Constraint-Based Tutors: What we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, 22(1-2), 39-72 (2012).
10. Weerasinghe, A., Mitrovic, A., Thomson, D., Mogin, P., Martin, B.: Evaluating a General Model of Adaptive Tutorial Dialogues. Proc. AIED, pp. 394-402 (2011).
11. Mitrovic, A., Mathews, M., Holland, J. Exploring two strategies for teaching procedures. S. Cerri, B. Clancey (Eds.) Proc. ITS 2012, LCNS 7315, pp. 502-507 (2012).
12. Elmasri, R., Navathe, S. B. *Fundamentals of database systems*. Addison-Wesley (2010).
13. Phillip, G. C. Teaching database modeling and design: areas of confusion and helpful hints, *Journal of Information Technology Education*, 6, 481-497 (2007).
14. Weerasinghe, A., Mitrovic, A., Martin, B.: Towards individualized dialogue support for ill-defined domains. *Artificial Intelligence in Education*, 19(4), 357-379 (2009).
15. Ohlsson, S.: Learning from Performance Errors. *Psychological Review*, 103(2), 241-262 (1996).